

# Design and Application of A Development Framework Based-on Shared Farm Digital Control Platform

Xi Liya<sup>1</sup>, Pen Yan<sup>1\*</sup>

<sup>1</sup>Institute of information science and engineering, Wuchang Shouyi University, 430064, Hubei

**Abstract.** Agriculture through science and technology is an effective way to promote the development of agricultural economy. In response to the problem of the shortage of urban arable land and the barrenness of rural land, it is combined with the sharing economy and agriculture through the Internet platform to propose the concept of using idle rural land to realize urban people's online pastoral life. Based on this, the application research on the integration of agriculture and the Internet has been carried out. Taking the needs of shared farm management as an example, through the analysis and research of Nginx, Flask and FFmpeg technologies, the development framework of NFF (Nginx+Flask+FFmpeg) web server with live broadcast function is designed. Based on the NFF framework, the management about land use patterns and planting methods in shared farms have been realized. Real-time live broadcast of farms has also been realized, which establishes a connection between consumers, growers and managers, and draws closer The distance between consumers and farms. The article focuses on the design ideas and application methods of the NFF framework in the agricultural management system, and the application of Blueprint technology in the live broadcast module. From the perspective of the application of the shared farm management system, the NFF framework can ensure the stability and scalability of the system functions, while also reducing the difficulty of development and improving the efficiency of development. It provides design ideas and application methods for Internet technology in agriculture. It is helpful to promote the sustainable development of the concept of "Internet + Sharing", provides a stage for the sharing economy to drive agricultural development, and provides a reference for the structure of a new agricultural management system.

## 1 Page layout

Hubei is known as the "Province of Thousand Lakes, Land of Fish and Rice", and is a major agricultural province in the central region. The use of the Internet to improve agricultural production, operation, management and service levels and to build a new agricultural management system is an effective way to promote the development of agricultural economy [1].

With the continuous advancement of urbanization, the urban population has increased and the arable land has decreased. However, most of the rural labor force goes out to work, and the lack of labor force leads to the barrenness of the land [2]. In order to satisfy the urban population's yearning for rural life and green organic food [3], and alleviate the contradiction between the lack of urban arable land and the barrenness of rural land, guided by the design concept of "Internet + agriculture + sharing" [4], it designs an intelligent management system for online shared farms based on cloud platform. The system adopts a mode of online and offline integration. It manages to the usage mode of rural idle land and planting method of crop through the online, which realizes the online pastoral life of urban people.

## 2 Analysis and research of development architecture

The system mainly includes server, user terminal (including PC terminal, mobile terminal), live broadcast server. Users can lease land, buy seeds, select planting trusteeship models online. And the crop growth is presented to the user through live webcast.

The system structure is shown in Figure 1.

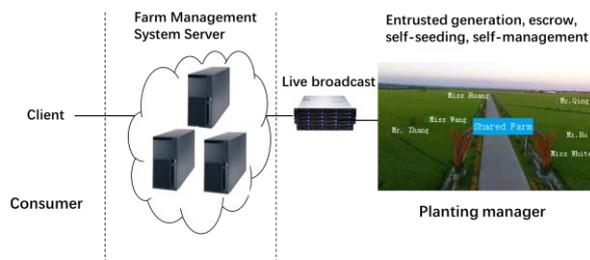


Fig. 1. System Structure Diagram

\* Corresponding author: [pey9076@qq.com](mailto:pey9076@qq.com)

## 2.1. Nginx+Flask framework

Considering the system's cross-platform, stability and other performance requirements, as well as the effectiveness of the development cycle, the Nginx+Flask system development framework [5]-[6] is used, as shown in Figure 2.

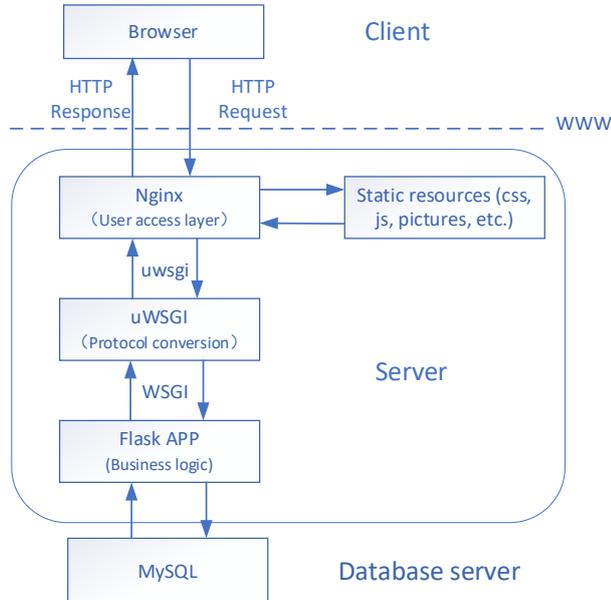


Fig. 2. Nginx+Flask Framework

Nginx is a free, open source, high-performance HTTP server that can run on Linux, Windows and other operating systems. It is with cross-platform features. Because Nginx uses an event-driven architecture, it can support millions of TCP connections, can better support large business volumes, achieve load balancing, and have good stability. Nginx has excellent static content processing capabilities, but it has weak processing capabilities for dynamic content. Therefore, in order to achieve a better client response, Nginx forwards dynamic content to the uWSGI server. The uWSGI server implements the wsgi protocol specification, and communicates with the Nginx server through the uwsgi protocol.

Flask is a lightweight web application framework written in Python, which can be well integrated with the MVC model, and can complete feature-rich small and medium-sized websites or web services in a short time, with high development efficiency [7]-[8]. Flask is a wsgi application, which is implemented by the wsgi protocol. So the WSGI protocol is used between the uWSGI server and the Flask application. Flask APP is designed and developed according to the RESTful specification. The front end sends requests according to a predefined grammatical format. The server does not need to parse various requests as before. It not only reduces the server's development workload and improves development efficiency. And the architecture platform provides unified and standardized services to meet the needs of multi-platform data sharing and interaction.

## 2.2 Nginx+ FFmpeg framework

FFmpeg is not only an audio and video codec tool, but also a set of audio and video codec development kits. It provides developers with a rich audio and video call interface, which can be used as a core video player, a core Directshow Filter, and a core transcoding tool. The audio and video codec functions are rich, which can well meet the audio and video development. The system involves the function of crop monitoring video live broadcast, and needs to obtain the real-time monitoring video of the network camera and transmit it to the user. To solve this problem, FFmpeg+Nginx is used to build a video streaming server to realize the monitoring video streaming of the network camera. The user page uses a player to realize the real-time monitoring of the live video streaming display.

## 3 System overall design

According to system requirements and function division, the overall system development architecture based on Nginx+Flask+FFmpeg is designed, as shown in Figure 3.

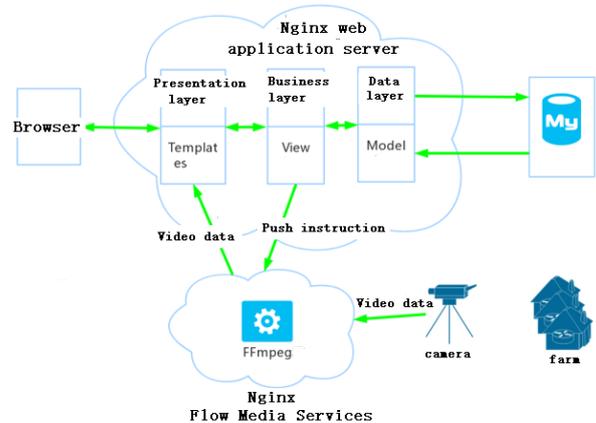


Fig. 3. System Overall Architecture Diagram

### 3.1 System server

The system server is running on Ubuntu 16.04 system, using Nginx as the web server to receive and parse the user's URL request. The server application uses the Flask framework, based on the RESTful standard, and is developed with Python language in accordance with the MVC three-tier architecture. The presentation layer is designed with multiple Templates (Jinja2 templates) to render web pages, and the business layer is designed with views to implement business logic and data. Data layer is designed to access with Model to MySQL database.

Based on the Flask framework, the browser initiates a request to the server of the system (here, the Nginx server). This request matches the corresponding URL in the URL mapping of the program, and then accesses the corresponding view function. If the data is not involved to be called, the view function returns a template for rendering and sends it to the client [5]. If the data is needed to be called, then the view function calls the

model class to find the data in the database, and then returns step by step. The view function fills the returned data into the web page template through the Jinja2 template engine, and finally returns the web page to the browser, returning the data [9].

### 3.2 Flow media services

When realizing the live broadcast function, you need to perform to push the stream operation about camera in the streaming media server. When there are more cameras to push the stream, it will put pressure on the server, affect the running speed of the server, and affect the user's access experience. Therefore, it is necessary to separate the streaming media server from the system server, and construct the streaming media server with Nginx+FFmpeg. While the FFmpeg obtains video data from the camera, the user can pull the video data through the web player. The video data which is pulled will be played on the web page, so as to realize the real-time live broadcast of the land.

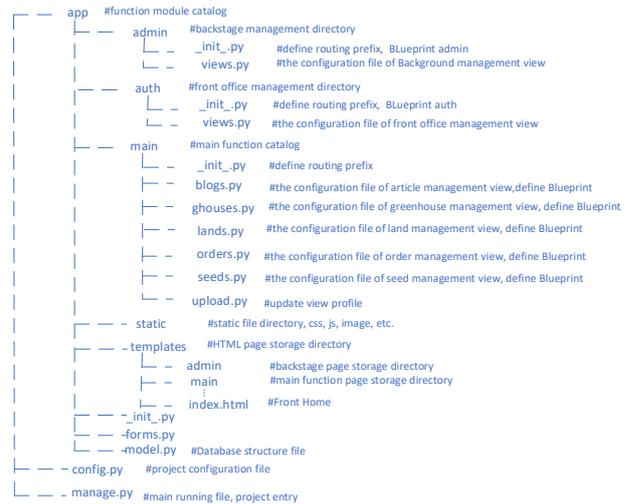
## 4 System implementation

The system is composed of two parts: the foreground and the background. The foreground is for consumers and the background is for administrators. It realizes functions such as user management, commodity management, shopping cart, order management, video monitoring, article publishing, and user comments [10].

### 4.1 Application of Flask blueprint in the system

Blueprint is the function of implementing modular applications in the Flask architecture. Simply put, it is a container for storing operation routing mapping methods, which is mainly used to realize the correlation between client requests and URLs. It divides into different categories according to different functions, reduces the coupling degree between various functional modules, and realizes the modularization of code. It can be shared not only within one application but also across multiple projects.

In this system, Blueprint technology is adopted for design and development to realize code componentization. According to the system functions, it divides separately the foreground modules and background modules. The front and back functions are managed separately to make the code clearer. When troubleshooting, you can find the corresponding code quickly according to the page routing. Each module is a collection of the same function. The system has designed 7 Blueprints, namely auth, admin, seed, land, blog, gate and order. The Flask modular structure based on the blueprint design is shown in Figure



**Fig.4.** Modular Structure based on Flask

manage.py is the project entry file, system functions are concentrated in the app directory. config.py is the project configuration file, including database configuration information. In the app directory, the front-end (auth directory) and back-end (admin directory) functions is separated, and the main function is placed in the main directory. static is the static file directory, which stores css, js and image files. HTML pages are centralized in templates, and model.py is the database structure file.

### 4.2 Implementation of Video Monitoring Module based on Blueprint

To realize the farm live broadcast function, the administrator (backstage) needs to set IP address of the camera corresponding land first. After setting, the system will send commands to the streaming media server, and the FFmpeg will push the stream of video data. When users want to view the land which has been purchased by themselves, it can pull the stream through the player on the web page to realize real-time video playback.

The following takes the video live broadcast module as an example to introduce the realization of system functions with blueprints.

The administrator finds the add\_land() view function through the PSOT request of URL /land/add/, stores the land information in the database, sends a push instruction to the streaming media server, and turns on the camera to live broadcast.

Defines of the Blueprint:

```
admin = Blueprint('admin', __name__)
```

Registration of the Blueprint :

```
app.register_blueprint(admin, url_prefix ='/admin')
```

After defining the Blueprint and registering the Blueprint, you can directly call the add\_land() view function through the Blueprint. Its Blueprint application code show as below:

```
@admin.route('/land/add/', methods=['GET', 'POST'])
```

```
def add_land():
```

```
    form = LandForm()
```

```
    if form.validate_on_submit():
```

```
        data = form.data
```

```
land_count = Land.query.filter_by(number=data["number"],ghouse_id=data["ghouse"]).count()
if land_count > 0:
    flash("The land number in the greenhouse already exists! ", "err")
    return redirect(url_for("admin.add_land"))
land = Land(
    number = data["number"],
    prices = data["price"],
    area = data["area"],
    ghouse_id = int(data["ghouse"]),
    ip = data["ip"],
)
db.session.add(land)
db.session.commit()
flash("Add land information successfully! ", "ok")
return redirect(url_for("admin.list_land",page=1))
return render_template('admin/add_land.html', form = form)
```

Next, the user finds the corresponding `get_land(id)` view function through the GET method request of URL `/video/<int:id>/`, obtains the information of land about the id's from the database, loads the T5 player in the template and play the video from the camera.

Defines of the Blueprint:

```
land = Blueprint('land', __name__)
```

Registration of the Blueprint :

```
app.register_blueprint(land, url_prefix ='/land')
```

After defining the Blueprint and registering the Blueprint, you can directly call the `get_land()` view function through the blueprint. The code is as follows:

```
@land.route('/lvideo/<int:id>/', methods=['GET', 'POST'])
def get_land():
```

```
    try:
        land = Land.query.get(id)
    except Exception as e:
        print(e)
    return render_template('main/video.html', land = land)
```

The live broadcast effect is shown in Figure 5.



**Fig.5.** Live Effect

## 5 Conclusion

The system is developed based on the NFF framework. Judging from the load access and stress test of this system, the operation is stable.

Nginx+Flask+FFmpeg technology can not only ensure the stability and scalability of system functions, but also reduce the difficulty of development and

improve development efficiency. Have a certain reference.

The research and application of this project are in line with the overall deployment of the country to promote agricultural development. The platform's design ideas, design methods, design frameworks and research issues reflect the important role of information technology represented by the Internet in the development of the industry. It helps promote the sustainable development of the "Internet + Sharing" concept, provides a stage for the sharing economy to drive agricultural development, and provides a reference for the construction of a new agricultural management system.

## Acknowledgments

This article is one of the phased achievements of the Hubei Provincial Department of Education's scientific research project "Research and Implementation of a Digital Management and Control System for Shared Farms Based on Cloud Platforms" (project number: B2018348, document number: Hubei educational letter [2018] No. 9).

## References

1. Wan B. (2015) My country's rural areas will face another major change-"Internet + Sannong" research and thinking. *Agricultural economic issues*: 36: 4-7.
2. Liu S. (2016) Research on the problems and countermeasures of Hubei agricultural economic development. *Central China Normal University, Wuhan*.
3. An H, Chen M. (2005) Industrialization, urbanization process and the path choice of urbanization in my country. *Nankai Economic Research*,1:48-54.
4. Yang Q, Peng Y. (2015) "Internet + agriculture" is a concept or a way out. *Rural areas. agriculture. Farmers (version B)*,7:45.
5. Wu T. (2016) Design and implementation of material management system based on Flask framework. *Nanjing University, Nanjing*.
6. Liu L, Zhang G. (2017) Software design of smart home service system based on Flask. *Industrial control computer.*, 30: 109-110.
7. Grinberg, Miguel. (2014) *Flask Web Development:Developing Web Applications with Python*. O'Reilly Media,Sebastopol.
8. Lei L, E X, Yang F. (2016) The design and implementation of the educational administration system based on the open source framework Flask. *Information and Computer: Theory Edition*, 20:107-109.
9. Chen F. (2014) Design of Farm Production Management Information System. *Central China Normal University, Wuhan*.

10. Ding J. ( 2018 ) Analysis of the role of agricultural economic management on rural economic development. *Agricultural development and equipment*, 4:24-26.